



#### Moving Table (ADVANCE) GradWarp correction

- The ADVANCE moving table acquisition as proposed by Kruger et al. acquires a 2 or 3D data set while the table is moving along the frequency-encoding direction.
- If  $k_f$  is the frequency encoding direction,  $k_p$  is phase-encoding, and  $k_s$  is slice-encoding, then  $k_f$  is along the direction of table movement.
- GradWarp corrects for the gradient non-linearities by calculating the physical location of every pixel in an image and then applying an interpolation kernel to map the pixel in the image to a corrected image.
- The current algorithm assumes that there is a one-to-one mapping between the object being imaged and the location in the magnet. In other words each point in the object corresponds to a unique point in the magnet.
- When the table is moving this is no longer the case, as the object moves through the magnet the same point is imaged at different locations in the magnet. For this reason conventional 2D/3D GradWarp correction will not work.
- Recognize that for each line of k-space that is acquired,  $S_n(k_f, k_p(n))$ , it is acquired at the same physical location in the magnet, so in theory the correction should be the same for each line.
- GradWarp requires a 2D image on which to perform the correction. To do this we calculate a separate 2D/3D image for each line of k-space, calculate GradWarp on this image and then sum it with all of the other images generated from each different k-space line.

BEST AVAILABLE COPY

Here are the steps:

1. Each line of data is collected corresponding to  $S_n(k_f, k_p(n))$  where  $n$  is the view number in the acquisition. Typically this data is Fourier transformed along the  $k_f$  direction. Instead we are going to do a 2D FFT. Since  $k_p(n)$  is really a single value or a delta function for each  $n$ . This really corresponds to a 1D transformation modulation by a sin and cos or  $S'_n(f)e^{jk_p(n)p}$  where  $f$  and  $p$  are spatial coordinates.  $S'_n(f)$  will have values for pixels from  $m_f = 0$  to  $M_f - 1$  and  $e^{jk_p(n)p}$  should be evaluated from  $m_p = 0$  to  $M_p - 1$  where  $k_p(n) = n_p \Delta k_p = m_p / FOV_p$ .
2. 2D GradWarp is calculated on each of the  $n$  images generated in step one. Note that the same physical GradWarp corner points can be used and the same physical mapping of unwarped image location to warped image location can also be used. This means that the GradWarp correction coefficients can be precalculated for all of the pixel in the uncorrected image.
3.  $\tilde{S}_n$  is now the corrected image after GradWarp. These images need to be corrected for table position as in the original technique, where  $S''_n(f, p) = S'_n(f - Z(n), p)$  and  $Z(n)$  corresponds to the physical table location for the  $n$ th line of k-space that is acquired. Unfortunately since we are already in the image domain it is not possible to correct for the sub-pixel registration using phase modulation. Instead it probably makes more

sense to combine step 2 and 3 and subtract  $Z(n)$  as part of the GradWarp pixel registration procedure. In one step this would remap the pixel to the correct physical magnet location.

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKewed/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**